UCRL-ID-118894

Recommendations for Protecting

National Library of Medicine

Computing and Networking Resources

Richard Feingold

510.422.1789
FAX 510.423.8002
Email feingoldra@llnl.gov

Secure Systems Services
Computer Security Technology Center
Lawrence Livermore National Laboratory

November 1994

Work performed under the auspices of the U. S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Introduction

Protecting Information Technology (IT) involves a number of interrelated factors. These include mission, available resources, technologies, existing policies and procedures, internal culture, contemporary threats, and strategic enterprise direction. In the face of this formidable list, a structured approach provides cost effective actions that allow the organization to manage its risks.

We face fundamental challenges that will persist for at least the next several years. It is difficult if not impossible to precisely quantify risk. IT threats and vulnerabilities change rapidly and continually. Limited organizational resources combined with mission restraints—such as availability and connectivity requirements—will insure that most systems will not be absolutely secure (if such security were even possible). In short, there is no technical (or administrative) "silver bullet." Protection is employing a stratified series of recommendations, matching protection levels against information sensitivities.

Adaptive and flexible risk management is the key to effective protection of IT resources. The cost of the protection must be kept less than the expected loss, and one must take into account that an adversary will not expend more to attack a resource than the value of its compromise to that adversary. Notwithstanding the difficulty if not impossibility to precisely quantify risk, the aforementioned allows us to avoid the trap of choosing a course of action simply because "it's safer" or ignoring an area because no one had explored its potential risk.

Recommendations for protecting IT resources begins with discussing contemporary threats and vulnerabilities, and then procedures from general to specific preventive measures. From a risk management perspective, it is imperative to understand that today, the vast majority of threats are against UNIX hosts connected to the Internet.

Threats and vulnerabilities

IT Security threats (the potential for adverse action against a resource) and vulnerabilities (weaknesses that may be exploited by an adversary) change rapidly over time. To make the most effective use of our limited resources, we need to concentrate on those vulnerabilities being exploited today, prepare for those that seem likely to be soon exploited, and keep a watchful eye on the theoretically possible. Of course, clever adversaries could recognize this approach and switch to new methods as we stymie the old ones. What we've found, however, is that most attacks are copy-cat rather than original. When an attack fails at one location, most attackers move on to another location rather than change methodology.

(This last observation is a classic example of "crime displacement." When one community gets tough on crime, the criminals move to a new, easier neighborhood. In today's environment, not only do security professionals usually have to accept displacement rather than being able to stop the activity, but they must also contend with the reverse issue. If your location's security is weaker than another, it will likely become a target. Further, once you become a target of successful attacks, you will become popular quickly, and continue to be the target for quite a while, even if you strengthen your defenses.)

This assessment will address the threats and vulnerabilities of outsiders remotely exploiting insecure UNIX workstations via the Internet and telephone access. (There are other threats and vulnerabilities outside the scope of this assessment, such as insiders taking advantage of their position and special knowledge and viruses for PC's and Macintoshes.) Once outsiders obtain access to UNIX workstations, they frequently obtain privileged access on one or more workstations on a Local Area Network (LAN). They then install monitoring programs ("sniffers") to obtain the initial transactions of remote log on and file transfer sessions, capturing many host name / user name / password combinations as well as other access information. It's important to recognize that this immediately extends the vulnerability to all systems on the LAN that have reusable[1] password authentication or exchange sensitive information.

There are many ways for intruders to gain initial access to UNIX workstations. They may exploit configuration weaknesses to obtain password files and then

[1] "Reusable" passwords are the traditional kind that users may change periodically, but remain the same in between changes. "Non-reusable" passwords are those that change after each use, either because they are a function of time, or the function of a random and always changing challenge.

run an enhanced dictionary search against the file; they may exploit configuration weaknesses to obtain trusted access through remote log in capabilities (the “r” commands); they may exploit configuration weaknesses to add or alter entries in password files; they may exploit unpatched vulnerabilities in operating systems.

Once intruders gain privileged (or root) access, there is little they cannot do. They may install monitoring programs; they may create new or change existing accounts; they may substitute Trojan Horse[2] software for legitimate systems, disguising their activities and / or leaving themselves backdoors for subsequent re-entry; they may launch attacks against other locations, causing you potential liability.

[2]Trojan Horse programs like their counterpart in Greek Mythology, appear to be benign or beneficial while they carry a surreptitious and usually malicious agenda.

Preventive measures

The National Library of Medicine (NLM) must prevent the unauthorized access of all accounts, the unauthorized increase of account privileges, the revelation of sensitive information, the alteration of information, and do so while not compromising the availability or quality of its services.

Good security begins with good backups. The schedule must be commensurate with the value and time needed to create the information. Store backups securely and physically separate from the system. Keep a log of your backups, and keep the backups, themselves, for a long time. Most importantly, test the backups periodically.

We will now look at the special case of telephone access and then the specific NLM environment, breaking it into four categories. We will discuss each in turn, comprehensively describing how one safely connects to the Internet.

Secure telephone access

The telephone allows world wide access to any network that has dial in modems. Intruders routinely compromise telephone systems for two primary goals: (1) free service; (2) the ability to manipulate certain switching functions. These intruders are not nearly as common[3] as the Internet intruders. They are most interested in obtaining free telephone access for themselves or to sell to other criminals who then resell these services for quick, lucrative profits. The Secret Service estimates that this activity, technically known as "toll fraud," was a three to four billion dollar criminal activity in 1993.

Modems used for inbound access on terminal servers and PC or timesharing systems are an obvious risk and need appropriate controls. We recommend keeping the modem telephone numbers confidential even as we recognize that such protective measures are usually of limited effectiveness.[4] Even truly confidential numbers are discovered since intruders frequently use PC's to automatically dial every number in a telephone exchange, testing for modems.

Dial back[5] modems have value, but it's important to recognize their limitations. Call forwarding and other attacks against telephone switches defeat dial back modems; but note that these kinds of attacks are significantly less frequent than Internet UNIX attacks—today. Dial back is also fairly inflexible when it comes to travel. Protect dial in modems with strong, frequently changed individual or non-reusable passwords. Shared passwords are usually a bad idea. Not only do they confound accountability, but they are never really secret. Strictly control modems on PC's since they may allow incoming access and the PC's usually lack any kind of authentication. An attacker can easily use PC on a local area network, such as Ethernet, to passively monitor its entire LAN segment.

---

[3]Unfortunately, "Phone Phreaking" as it's called is a relatively mature form of IT crime, and those who engage in it are rather sophisticated.

[4]So called "security through obscurity" rarely works against determined attackers for reasons that go beyond the scope of this document. This is not to say by any means that you shouldn't bother to limit knowledge of sensitive information—just don't rely on such limitation as the primary mode of security.

[5]That is, modems programmed to only dial specific telephone numbers depending on the identification they receive.

The specific NLM environment

The NLM IT environment consists of the following:

UNIX workstations:
- Sun workstations
- SGI workstations
- ULTRIX workstations
- HP workstations

Non-UNIX mainframe:
- IBM Mainframe / 3090J

Non-UNIX workstations:
- DEC alpha / Windows NT
- PC's
- Some Macintoshes

Networks:
- Novell LAN
- Gator boxes
- Ethernet LAN
- Internet connections

UNIX workstations

UNIX workstations connected to the Internet represent the vast majority of vulnerabilities in today's environment. The recommendations below in "Connecting securely to the Internet" detail how to protect these hosts.

Non-UNIX mainframe

Intruders have not targeted IBM mainframes in the sense of discovering and exploiting their vulnerabilities. This is not to say that they won't—or can't. Today's mainframe threat is the capture of passwords and other access information on a Local Area Network by a monitor installed on a compromised UNIX workstation. Note that intruders may exploit improperly protected FTP

services provided by the mainframe, but this would be unlikely to lead to further compromise of host.

Apply the password and account management advice in below to the IBM. Use non-reusable[6] passwords for all sensitive accounts. These are time dependent or challenge/response systems and are discussed in the "Connecting securely to the Internet" section below. Encrypt sensitive transmissions over any LAN segment exposed to the Internet. Consider encrypting locally stored information of a highly sensitive nature. Initiate or actively maintain education and awareness activities—users of mainframes tend to be isolated and feel overly secure.

If providing FTP services, do not allow write or delete access on outgoing files and directories. Do not allow read or delete access to any incoming repositories. While this latter constraint may not seem significant, improperly protected incoming areas on FTP servers have been used as exchange points for pirated software and pornographic materials—at great embarrassment and potential liability to the victim site.

Non-UNIX workstations

The models for non-UNIX workstations are the single user PC (MS/PC-DOS or Macintosh) and the various servers. Their vulnerabilities have increased as they've become increasingly networked. Fortunately, LAN file sharing protocols such as IPX (Novell), LAN Manager (Microsoft), and so on, are not usually carried on the Internet. Naturally, there is a threat if intruders compromise the Local Area Network workstations. However, if the workstations are providing FTP services, then the concerns expressed above in the mainframe discussion also apply. Note, for example, if one enables FTP for a Macintosh using NCSA/BYU telnet (say for downloading a file) it also automatically enables incoming FTP. If the operator had failed to establish and set a password, a clever intruder could literally take complete control of all the Macintosh file systems—reading or copying, deleting, writing, and replacing existing software. Verify that such capabilities are turned off or very carefully controlled. It is my understanding that no non-UNIX workstations at NLM are providing NFS services. If that's not the case, additional preventive measures would need to be taken.

[6]One time time dependent or challenge/response, discussed in a footnote 1 above.

Networks

Networks are usually the access path, not the direct target of attacks, with the exception of routers. At present, attacks against network protocols (other than TCP/IP and UDP used on UNIX workstations) are practically of no concern, especially since the Internet does not transport those protocols.[7] Manage router passwords as described below. Avoid performing router management functions that involve using passwords over the LAN. If possibly, use directly connected terminals.

[7]Other protocols can be carried on the Internet through a technique known as “tunneling.”

Connecting securely to the Internet

Acknowledgment

This section was adapted from a portion of the slides prepared by Steve Weeber of the DOE Computer Incident Advisory Capability for his excellent course "Connecting to the Internet Securely."

Securing the host

The following subsections discuss system security enhancements related to patches, authentication, account management, file ownership, file integrity, accounting, and configuration checks for UNIX hosts.

Patches

Vulnerabilities and scripts to exploit them are regularly being posted to mailing lists and newsgroups. Subscribe to the lists bugtraq (send email to `bugtraq-request@fc.net` with body of message `subscribe your-email-address`) and 8lgm (send email to `8lgm-request@bagpuss.demon.co.uk` with body of message `subscribe your-email-address`). Read the USEnet newsgroups such as `alt.security`, `comp.security.misc` (see Electronic Resources reference for others). Keep a detailed log of installed patches.

Obtaining Patches

From **Sun**:

- Local Sun Answer Center
- Anonymous FTP at `sunsolve1.sun.com`
- WWW at `http://www.sun.com/`
- Subscribe to Customer Warning System by sending email to
  `security-alert@sun.com`
  with body of message
  `subscribe CWS your-email-address`

From **HP**:

- WWW at http://support.mayfield.hp.com
- Join email response system by sending email to
  support@support.mayfield.hp.com
  with subject
  send guide.txt

From **SGI**:

- Anonymous FTP at ftp.sgi.com in the directory /security.

From **DEC**:

- Contact your Digital support channels.

Authentication, integrity, accountability, and secure configuration

Passwords

They're still important, even with sniffers. Use the largest password space possible—that is, don't limit yourself to any subset, such as same case alphabetic characters. *Pass-phrases* are convenient to memorize—using the first letter of each word, substituting the numerals "0" and "1" for the letters "o" and "i", respectively. "I like to sip soda in my sneakers" becomes 1ltssims. Or join two unrelated words with a non letter character. For example: Soda*Sneak or car=sponge.

Check for weak passwords using the Crack program (available via anonymous FTP from black.ox.ac.uk in /pub/security/crack41.tar.Z). This checks existing password files for weak choices, using a brute force dictionary search, information from the GECOS field, and configurable rules. There are some drawbacks. It is CPU intensive the first time; there is lag time between selection and detection, and you need a dictionary that's as big as the bad guy's.

If possible, install passwd+ (available via anonymous FTP from ftp.dartmouth.edu in /pub/security/passwd+.tar.Z). It forces users to choose "good" passwords. It has dictionaries, configurable rules, allowing enforcement of password policy with no CPU overhead and no lag time.

The UNIX password file /etc/passwd is world readable because many programs need its information. However, most programs do not need the password field. Thus, the operating system (OS) can be modified to move the password field to a "shadow" file readable only by the privileged account root.

Most recent OS releases include support for shadow passwords. If your OS does not, there is a public domain kit (available via anonymous FTP from `ftp.std.com` in `/src/util/shadow`).

Password aging limits the window of vulnerability for sniffed passwords, passwords that are being cracked, and accounts that are not being used. That is, the user must change their password periodically. The mechanism varies between vendors and operating systems (see the manual entry for passwd). In general, expire the password in some number of days with a selectable pre-expiration warning period. For example, for an ordinary user, you might expire a password in 90 days giving them 7 days warning. For a privileged user, say 30 days with a 7 day warning.

Passwords sent in clear text over networks have rapidly become the most popular attack mechanism. Consider non-reusable passwords. That is, either one time passwords that change with time (usually once a minute) or through generating a cryptographic sequence, or a challenge response mechanism. These passwords are of no use to an attacker, even if captured with monitor. There are software and hardware solutions. The public domain S/key software is available from `thumper.bellcore.com` in `/pub/skey`. There are several popular token (Smart Card) systems, such as SecureID and Enigma Logic. (See CIAC Advisory E-12 for more information.)

Account Management

Verify that all vendor supplied default accounts that support logins have strong passwords. Older operating systems and third party software systems are especially suspect.

Lock out accounts that do not support logins (but are needed for other purposes), by following these steps.

- Place an asterisk (`*`) in the password field in `/etc/passwd` or the shadow file
- Change the login shell to `/bin/false`
- Change the home directory to `/nodir`

Note that you cannot comment out a passwd entry. There is no comment character, and a `#` in column 1 becomes the first character of the username.

Dormant accounts are a prime target for intruders. Periodically expire unused accounts automatically if the system allows it—manually if necessary.

Avoid shared (group) accounts. Use group access permissions to share common files. If more than one person needs the same home directory, use multiple user

names rather than sharing the account and password. Do not share privileged root accounts; rather, create distinct privileged accounts (user 0, group 1).

Carefully control access to root. Do not allow direct root logins, especially over the network. Remove the `secure` keyword from `/etc/ttys` or `/etc/ttytab`, so those devices will not support a privileged login. This will force users to login to an unprivileged account and then `su` to `root`. Carefully control the number users allowed to do that, and remember to add them to the `wheel` group.

File ownership

Programs and directories used by `root` should be owned by `root`, not `bin`. Make certain Sun systems have patch 100103 installed. Root's search path should not include directories it does not own, since that might cause it to execute Trojan Horse substitute programs with root privilege. Nor should root's search path include "." (current directory) for the same reason. The public domain system `tiger` (available via anonymous FTP from `net.tamu.edu` in `pub/security/TAMU`) performs an exhaustive search of programs that may be executed by root and reports any problems.

Device ownership

Inappropriate device ownership allows intruders to eavesdrop and/or manipulate input to the host. The following devices should be owned by the user logged into the console: `framebuffer`, `keyboard`, `mouse`, `microphone`. See, for example, `/etc/logindevperm` under Solaris 2.x and `/etc/fbtab` under SunOS 4.x.

File integrity

Intruder Trojans are increasingly common and nearly impossible to detect. They may have the same access, modification, and i-node change time as the original, the same file size, and the same /bin/sum checksum. We thus need strong authentication, including cryptographic checksums.[8]

The Security Profile Inspector (SPI) is a very effective general purpose product. It is free to sponsoring government agencies and available at nominal charge to all

[8]The standard /bin/sum check was designed to catch unintentional changes (such as transmission noise). It can be easily defeated by a determined adversary. Cryptographic checksums cannot be.

others (see footnote[9]). Its Binary Authentication Tool provides the ability to determine both system object authenticity and patch currency. Tripwire (available via anonymous FTP from `coast.cs.purdue.edu` in `/pub/COAST/Tripwire`) is a standalone tool for maintaining a database of file checksums, supporting up to 9 different checksums per file.

Accounting

Process level accounting (`pacct`) is usually turned off by default on most UNIX systems. It provides a history of all commands executed on the system, but must be balanced against CPU and storage overhead. The `lastcomm` command allows you to review the processes executed selected by command, user, and terminal name. See `man` page for `acct(8)` for more information.

Configuration checks

Automated tools for checking the security of system configurations include SPI, Tiger, COPS, and Securscan. In general they check file ownerships, known vulnerabilities, weak configurations, and patch status.

SPI, mentioned above, is the recommended product (see footnote for details to acquire it). It is the most comprehensive and is actively and continually maintained under sponsorship of multiple U.S. government agencies. It is the ongoing work of the Computer Security Technology Center at Lawrence Livermore National Laboratory, under the project management of Tony Bartoletti (510.422.3881, email `azb@llnl.gov`). As already discussed, SPI also checks the status of system binaries.

COPS (Computer Oracle and Password System) is public domain software that is now several years out of date but still of some value. It is a configuration checking system written by Dan Farmer. It is available via anonymous FTP from `info.cert.org` in `/pub/tools`.

Tiger was written in response to widespread intrusions. It checks for known signs of intrusion, configuration vulnerabilities, unpatched vulnerabilities, and is

---

[9]U.S. Government agencies may purchase SPI through the Energy Science & Technology Software Center (ESTSC) for a nominal charge. Contact information:

Internet: ESTSC@ADONIS.OSTI.GOV
Mail: Energy Science and Technology Software Center
PO Box 1020
Oak Ridge, TN 37831-1020
Phone: 615-576-2606; Fax: 615-576-2865 (verification 615-576-2606)

being actively maintained. It includes the `tigexp` utility to explain every warning message issued. It is available via anonymous FTP from `net.tamu.edu` in `/pub/security/TAMU`.

Securscan checks for known vulnerabilities in SGI IRIX. These include permissions, NIS, and X-Windows. It incorporates the Computer Emergency Response Team (CERT) and Computer Incident Advisory Capability (CIAC) advisories. It is actively maintained, and available via anonymous FTP from `ftp.vis.colostate.edu` in `/pub/irix/security`.

Network Controls

The following sections discus security enhancements related to securing the network—in contrast to the host. Topics include general philosophy; the Internet daemon—configuration, problem protocols, and TCP wrappers; the portmapper—operation and securelib; other network daemons; and firewalls.

Overall philosophy

Control what services you offer—`finger`, `rusers`, `systat`. Control who may connect to your services—local machines, local site, world. Control what information your systems will give out for free—login banners, `DNS HINFO`, `WWW`. For example, will a `telnet` to port 25 reveal your `sendmail` version?

The Internet daemon

Inetd listens on reserved ports and launches daemons in response to requests received. It is controlled by `/etc/inetd.conf`. The following table is an example (not a recommended) configuration.

Service name	Socket type	Prot-ocol	Wait status	Uid	Server program	Server arguments
ftp	stream	tcp	nowait	root	/usr/etc/in.ftpd	in.ftpd
telnet	stream	tcp	nowait	root	/usr/etc/in.telnetd	in.telnetd
shell	stream	tcp	nowait	root	/usr/etc/in.rshd	in.rshd
login	stream	tcp	nowait	root	/usr/etc/in.rlogind	in.rlogind
exec	stream	tcp	nowait	root	/usr/etc/in.rexecd	in.rexecd
comsat	dgram	udp	wait	root	/usr/etc/in.comsat	in.comsat
talk	dgram	udp	wait	root	/usr/etc/in.talkd	in.talkd

Make certain that this table is consistent with your operations.

Trivial File Transfer Protocol (TFTP)

TFTP has no authentication. It is used to boot diskless workstations, routers, etc. If incorrectly configured, it will transfer any file on the system, e.g., `/etc/passwd`. You can secure TFTP either by turning it off in `/etc/inetd.conf`, or using a version that can limit the retrievable files. For example, under SunOS, the server arguments field contains:

```
in.tftpd -s /tftpboot
```

If you require TFTP, use TCP wrappers (see subsection below for access information) to control which hosts can connect to it. If possible, block inbound TFTP packets at your Internet router (TCP port 69).

File Transfer Protocol (FTP)

Make certain that any PC workstation (Macintosh or DOS) with a TCP/IP package does not by default enable FTP with no authentication. If so, it then allows full access to local disks. For UNIX hosts, carefully configure anonymous FTP as follows.

- Create an ftp user in `/etc/passwd`.
- Lock the password.
- Create an invalid shell.
- Create FTP's home directory owned by `root` (not `ftp`).
- Set its protection mode to 555.
- All subdirectories under the FTP home must be owned by a user other than `ftp`.
- The `~ftp/etc/passwd` file should contain no encrypted passwords and as few usernames as possible. (See CIAC Bulletin D-19 for more information.)

Remote login and shell

This includes `rlogin` and `rsh` as well as `rcp` and `rdump`. The remote commands do not require the transmission of passwords over the network. They are based on trusted hosts in the global file `/etc/hosts.equiv` and local files `~/.rhosts`. The default configuration for some systems is a `+` in the `/etc/hosts.equiv` file. This allows access from remote machines if the remote username exists on that system. Secure the remote commands by:

- Carefully choosing the hosts in `hosts.equiv`
- Carefully monitoring the contents of users' `.rhosts` files—not allowing `+`
- Using TCP wrappers to limit the hosts allowed to connect
- Using the `logdaemon` package to replace `rlogind` and `rshd` with more secure versions—with extra logging and controls on `.rhosts` files

The `logdaemon` package is available via anonymous FTP from `coast.cs.purdue.edu` in `/pub/tools/unix/logdaemon`.

TCP wrappers

TCP wrappers perform access control and logging for daemons launched by `inetd`. The "wrapper" program is executed before the real daemon is launched. It can, on a per-daemon basis:

- Generate a log message
- Send mail to you
- `finger` the remote host
- And accept or deny the connection based on source address

It is available via anonymous FTP from `info.cert.org` in `/pub/tools/tcp_wrappers`.

The portmapper

The portmapper is a registrar for network daemons since they do not have fixed, well-known ports. It is generally used with RPC based services such as `keyserv`, `nfs`, `mountd`, and `bootparam`. The portmapper reveals potentially compromising information about a system concerning the NFS and NIS servers. If possible, block access to `portmap` at your Internet router (TCP & UDP port

111). (Note that a determined intruder can still exhaustively scan port ranges to locate desired daemons.)

Network File System (NFS)

The mount daemon allows file systems to be exported via NFS. It is controlled by the file `/etc/exports`. By default, file systems are exported insecurely with world read/write access and mountable by *anyone* on the Internet. You can secure NFS as follows:

- Use access lists.
- Export filesystems read-only whenever possible.
- Do not export filesystems to yourself, since a vulnerability in portmapper allows access to anyone.
- Run `fsirand` on your partitions periodically to randomize the NFS Inode generation numbers and make NFS file handles difficult to guess.
- If possible, block inbound NFS packets at your Internet router (UDP port 2049).

Network Information Service (NIS)

NIS manages the `/etc/passwd` and `/etc/group` databases for groups of hosts. The standard server will give out all databases to any requester that knows the NIS domain name. The latter is easily guessed—often by simply telneting to the mail port. If you don't need NIS, simply turn it off. On Suns, for example, use `resolv+` (available via anonymous FTP from `ftp.uu.net` in `/networking/ip/dns`) for Domain Name Service (DNS) and remove the startup lines for NIS in `/etc/rc.local`. If you do require NIS, use a server that can filter requests based on source address. On Suns, apply Patch 100482, and configure `/var/yp/securenets` with the proper address ranges.

sendmail

The software product `sendmail` is large, unattractive, antiquated, but unfortunately still very much in business. It runs as root and accepts connections from anywhere. Recent problems include intruders tricking it into running programs by bouncing messages with piped commands in the `From:` field, and another causing an overflow in the debug flag. You can secure `sendmail` by installing vendor patches as soon as they become available, switching to Berkeley sendmail v. 8 (available via anonymous FTP from `ftp.cs.berkeley.edu` in

/ucb/sendmail) with rapid fixes and full source available, or migrating to alternative mail products such as Smail or MMDF.

X-Windows

X was designed to allow easy sharing of resources—security was an afterthought. If it can display a window on your server[10], it can also monitor your keystrokes as well as insert keystrokes and events. There are three types of X access control: host-based using the xhost command to limit access to selected hosts; "MIT Magic Cookie" using a random number stored in ~/.Xauthority; and the XDM authorization, an encrypted extension of Magic Cookie for added security.

Gopher and Wide World Web

These new Internet services are extremely popular, easy to use, and provide vast amounts of new information. These servers distribute files to people who ask for them. You may want to limit who can receive what files. The servers may be tricked. For example, a gopher client may send /etc/passwd. Run server chroot'ed and as a non-privileged user. Gopher and WWW clients are partially controlled by the servers to which they connect. For example, UNIX NCSA Mosaic clients were vulnerable to malicious Uniform Resource Locators (URL's) on servers successfully telling the client to mail back /etc/passwd/. Run up-to-date clients, carefully choosing the actions that the client can take. A thorough treatment of servers is provided with the included reference "Securing Internet Servers."

[10]The terminology in X for client and server may seem anti-intuitive at first. The display system is the "server"; the (sometimes remote) application is the "client."

Conclusions

Successfully securing today's Information Technology resources combines specific knowledge with hard work and persistence. To paraphrase one of America's founding fathers, Thomas Jefferson: the price of security is eternal vigilance. For the most part, the "bad guys" out there have everything to gain and nothing to fear. Until lawmakers and law enforcement come up to speed, we're going to have to take an effective defensive posture and stay at least on par if not ahead of the intruders. The information provided in this and accompanying documents will give you the specific knowledge. Through our consultation and interactions we'll be able to help you most effectively focus your hard work.

Selected Bibliography

Cheswick, William R., and Bellovin, Steven M., *Firewalls and Internet Security,* Addison-Wesley, 1994.

[11]Department of Energy Computer Incident Advisory Capability (CIAC), Advisory E-12: Network Monitoring Attacks Update.

[12]Department of Energy Computer Incident Advisory Capability (CIAC), Bulletin D-19: Wide-spread Attacks on Anonymous FTP Servers.

[13] Department of Energy Computer Incident Advisory Capability (CIAC), *Securing Internet Information Servers,* CIAC-2308 R.0, UCRL-MA-118453, September, 1994.

[14] Feingold, Richard, *Electronic Resources for Security Related Information,* CIAC-2307 R.0, UCRL-ID-118613, November 1994.

Forum of Incident Response and Security Teams (FIRST), *SECURITY TOOLS AND TECHNIQUES Resource Library,* CD-ROM, October 1994.

Garfinkel, Simson, and Spafford, Gene, *Practical UNIX Security,* O'Reilly & Associates, 1991.

The Knightmare, *Secrets of a Super Hacker,* Loompanics, 1994.

[15] Pichnarczyk, Karyn, Weeber, Steve & Feingold, Richard, *UNIX Incident Guide: How to Detect an Intrusion,* CIAC-2305 R.0, UCRL-ID-118605, September, 1994.

Stang, David J., and Moon, Silvia, *Network Security Secrets,* IDG Books, 1993.

---

[11]In Appendix.
[12]In Appendix.
[13] Copy included with recommendations.
[14] Copy included with recommendations.
[15] Copy included with recommendations.

Appendix

CIAC Advisory E-12

and

CIAC Bulletin D-19

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551